

LAPORAN RESMI
PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK
PERCABANGAN DAN PERULANGAN



Fadilah Fahrul Hardiansyah S.ST., M. Kom

Ratri Maria Manik

3121600039

D4 TEKNIK INFORMATIKA – B

PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
TA 2022/2023

A. TUGAS PENDAHULUAN

1. Sebutkan dan jelaskan berbagai macam sintaks percabangan yang digunakan di Java!
 - a. If yaitu suatu percabangan yang digunakan ketika menginginkan suatu pernyataan itu dilakukan dengan syarat tertentu yang bernilai benar
 - Sintaks dari if.

```
if (ekspresi_boolean) {  
    Pernyataan1;  
}
```

- b. If-else yaitu suatu percabangan yang digunakan untuk mengeksekusi salah satu dari 2 pernyataan dari syarat tertentu yang pada if dapat bernilai benar atau salah.
 - Sintaks dari if-else

```
if (ekspresi_boolean) {  
    Pernyataan1;  
} else {  
    Pernyataan2;  
}
```

- c. Else-if yaitu suatu percabangan yang digunakan untuk memberikan kondisi tertentu pada bagian else.
 - Sintaks dari else-if

```
if (ekspresi_boolean1) {  
    Pernyataan1;  
} else if (ekspresi_boolean2) {  
    Pernyataan2;  
}
```

- d. Switch yaitu suatu percabangan yang digunakan ketika pemberian kondisi dengan beberapa syarat identik yang masing – masing mempunyai pernyataan yang berbeda – beda. Pada Java nilai yang dilewatkan pada switch harus bertipe int, short, byte, atau char.
 - Sintaks dari switch

```
switch (ekspresi) {  
    case nilai1: Pernyataan1;  
        break;  
    case nilai2: Pernyataan2;  
        break;  
    default: Pernyataan3;  
}
```

2. Sebutkan dan jelaskan berbagai macam sintaks perulangan yang digunakan di Java!
 - a. For yaitu sintaks perulangan yang digunakan saat sudah diketahui jumlah yang pasti dari perulangan tersebut.
 - Sintaks dari for

```
for (inisialisasi; kondisi; perubah) {  
    Pernyataan;  
}
```

- b. While yaitu sintaks perulangan yang digunakan saat belum diketahui pasti jumlah perulangan tersebut. Pernyataan while akan dikerjakan setelah pengecekan kondisi while bernilai true.

- Sintaks dari while

```
while (kondisi) {  
    Pernyataan;  
}
```

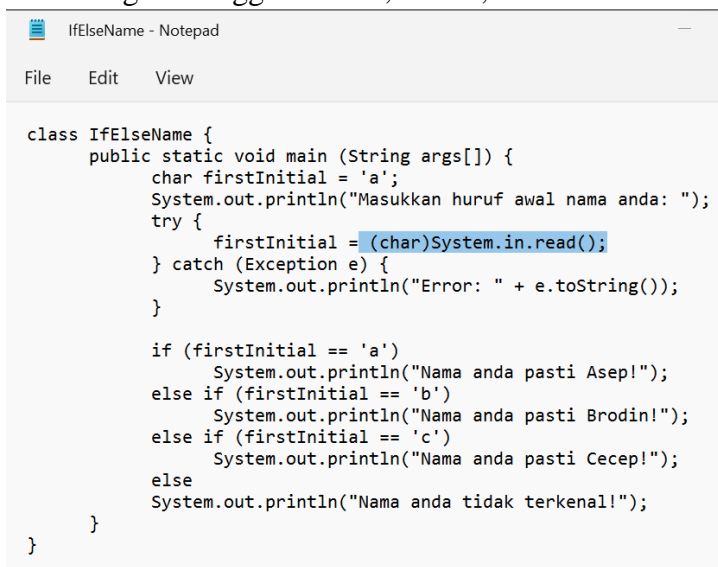
- c. Do-while yaitu sintaks perulangan yang digunakan saat belum diketahui pasti jumlah perulangan tersebut. Pernyataan “do” akan dikerjakan terlebih dahulu, lalu dilakukan pengecekan terhadap kondisi “while”.

- Sintaks dari do-while

```
do {  
    Pernyataan;  
} while (kondisi);
```

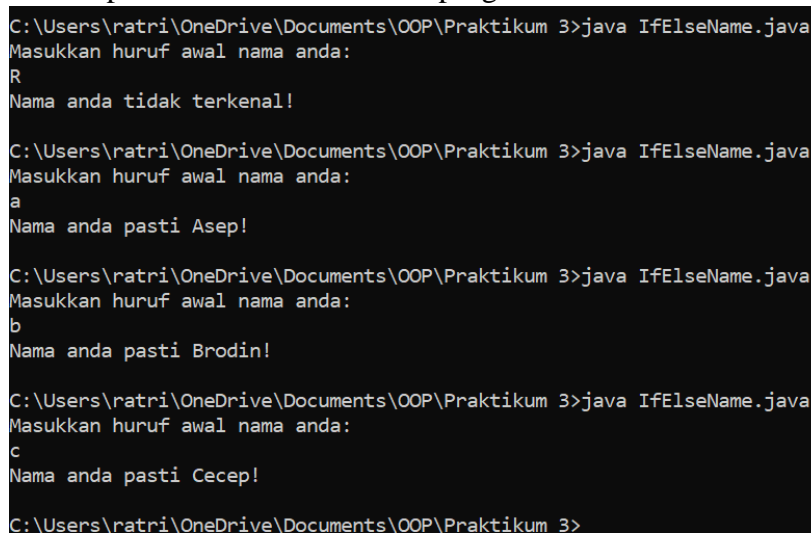
B. PERCOBAAN

1. Percabangan menggunakan if, if-else, else-if



```
class IfElseName {  
    public static void main (String args[]) {  
        char firstInitial = 'a';  
        System.out.println("Masukkan huruf awal nama anda: ");  
        try {  
            firstInitial = (char)System.in.read();  
        } catch (Exception e) {  
            System.out.println("Error: " + e.toString());  
        }  
  
        if (firstInitial == 'a')  
            System.out.println("Nama anda pasti Asep!");  
        else if (firstInitial == 'b')  
            System.out.println("Nama anda pasti Brodin!");  
        else if (firstInitial == 'c')  
            System.out.println("Nama anda pasti Cecep!");  
        else  
            System.out.println("Nama anda tidak terkenal!");  
    }  
}
```

Menampilkan hasil keluaran dari program di atas.

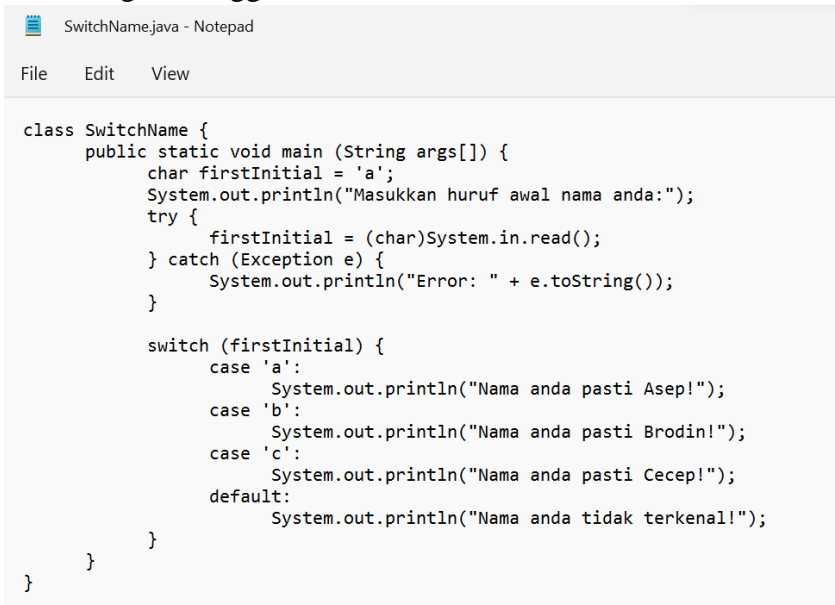


```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java IfElseName.java  
Masukkan huruf awal nama anda:  
R  
Nama anda tidak terkenal!  
  
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java IfElseName.java  
Masukkan huruf awal nama anda:  
a  
Nama anda pasti Asep!  
  
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java IfElseName.java  
Masukkan huruf awal nama anda:  
b  
Nama anda pasti Brodin!  
  
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java IfElseName.java  
Masukkan huruf awal nama anda:  
c  
Nama anda pasti Cecep!  
  
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>
```

Analisa :

Pada (if else if statement) jika ekspresi 1 tidak terpenuhi atau bernilai false, maka akan masuk ke ekspresi 2, begitu seterusnya hingga bernilai true, namun apabila program tersebut bernilai false untuk setiap ekspresi yang telah diproses, maka alur program akan menuju ke bagian else. Seperti pada program di atas, ketika diinputkan huruf "R", yang tidak memenuhi ekspresi 1, 2, atau 3, maka alur program tersebut akan menuju ke bagian else dan program berakhir. Namun ketika diinputkan huruf "a" yang memenuhi ekspresi 1 sehingga bernilai true, maka akan dieksekusi pada ekspresi 1 dan program berakhir, begitu juga ketika diinputkan huruf "b" dan "c" yang memenuhi ekspresi 2 dan 3.

2. Percabangan menggunakan switch

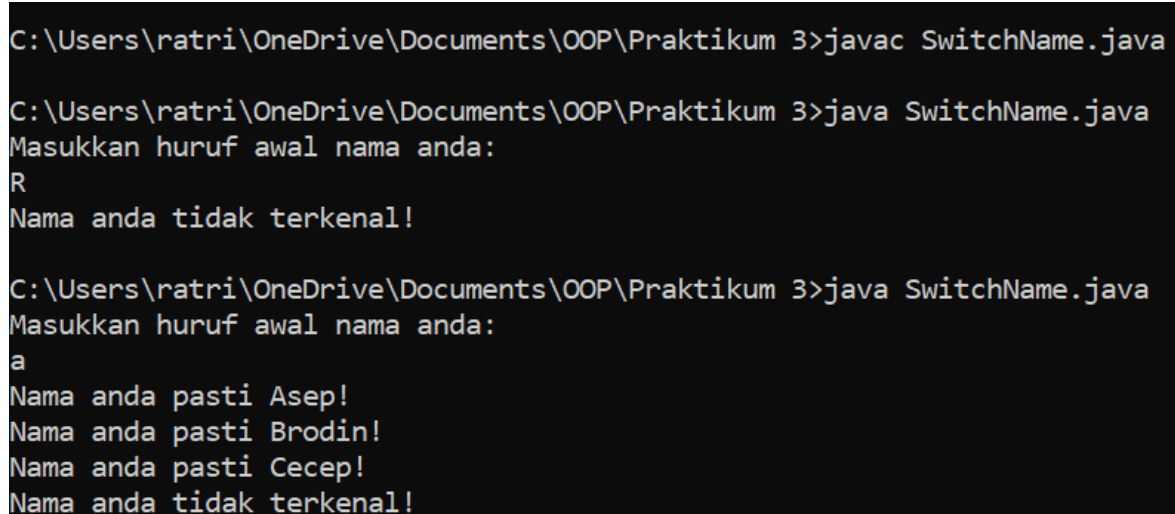


```
SwitchName.java - Notepad
File Edit View

class SwitchName {
    public static void main (String args[]) {
        char firstInitial = 'a';
        System.out.println("Masukkan huruf awal nama anda:");
        try {
            firstInitial = (char)System.in.read();
        } catch (Exception e) {
            System.out.println("Error: " + e.toString());
        }

        switch (firstInitial) {
            case 'a':
                System.out.println("Nama anda pasti Asep!");
            case 'b':
                System.out.println("Nama anda pasti Brodin!");
            case 'c':
                System.out.println("Nama anda pasti Cecep!");
            default:
                System.out.println("Nama anda tidak terkenal!");
        }
    }
}
```

Menampilkan hasil keluaran dari program di atas



```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>javac SwitchName.java

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java SwitchName.java
Masukkan huruf awal nama anda:
R
Nama anda tidak terkenal!

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java SwitchName.java
Masukkan huruf awal nama anda:
a
Nama anda pasti Asep!
Nama anda pasti Brodin!
Nama anda pasti Cecep!
Nama anda tidak terkenal!
```

Analisa :

Penggunaan percabangan menggunakan switch pada program di atas, ketika diinputkan suatu huruf yang tidak memenuhi ekspresi boolean sehingga bernilai false, maka alur program akan menuju ke bagian else. Selanjutnya jika diinputkan huruf "a" yang memenuhi ekspresi boolean sehingga bernilai true, maka ekspresi 1 dan ekspresi lainnya akan dieksekusi, begitu juga ketika diinputkan huruf "b" bernilai true pada ekspresi ke 2, maka ekspresi ke 2 dan ekspresi lainnya akan dieksekusi. Pengeksekusian terjadi pada tiap ekspresi dikarenakan tidak adanya penggunaan break pada program.

3. Percabangan menggunakan switch dengan break

```
SwitchNameBreak.java - Notepad
File Edit View

class SwitchNameBreak {
    public static void main (String args[]) {
        char firstInitial = 'a';
        System.out.println("Masukkan huruf awal nama anda:");
        try {
            firstInitial = (char)System.in.read();
        } catch (Exception e) {
            System.out.println("Error: " + e.toString());
        }

        switch (firstInitial) {
            case 'a':
                System.out.println("Nama anda pasti Asep!");
                break;
            case 'b':
                System.out.println("Nama anda pasti Brodin!");
                break;
            case 'c':
                System.out.println("Nama anda pasti Cecep!");
                break;
            default:
                System.out.println("Nama anda tidak terkenal!");
        }
    }
}
```

Menampilkan hasil keluaran dari program di atas.

```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java SwitchNameBreak.java
Masukkan huruf awal nama anda:
R
Nama anda tidak terkenal!

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java SwitchNameBreak.java
Masukkan huruf awal nama anda:
a
Nama anda pasti Asep!

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java SwitchNameBreak.java
Masukkan huruf awal nama anda:
b
Nama anda pasti Brodin!

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java SwitchNameBreak.java
Masukkan huruf awal nama anda:
c
Nama anda pasti Cecep!

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>_
```

Analisa :

Perbedaan penggunaan switch antara nomor 2 dan 3 yaitu terletak pada penggunaan break. **Break digunakan untuk mengakhiri sebuah eksekusi dalam statement, dengan menggunakan break maka pernyataan yang terdapat di dalam statement akan langsung berhenti.** Pada program di atas, ketika diinputkan huruf “R” ekspresi boolean tersebut bernilai false, alur program bernilai else, maka yang dieksekusi hanya pernyataan yang terdapat pada statement else tersebut. Ketika diinputkan huruf “a” ekspresi boolean bernilai true terhadap ekspresi 1, maka yang dieksekusi hanya ekspresi 1, dan program tersebut akan berhenti.

4. Perulangan menggunakan for

```
ForCount.java - Notepad
File Edit View

class ForCount {
    public static void main (String args[]) {
        int count=1;
        for (int i=0; i<9; i++) {
            for (int j=0; j<i+1; j++) {
                System.out.print(count);
            }
            count++;
            System.out.println();
        }
    }
}
```

Menampilkan hasil keluaran dari program di atas

```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>javac ForCount.java

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java ForCount.java
1
22
333
4444
55555
666666
7777777
88888888
999999999

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>_
```

Analisa :

Dapat dilihat pada program diatas menggunakan perulangan for yang sudah diketahui pasti jumlah perulangan yang akan diproses. Variable count pada awalnya bernilai 1, terdapat nested loop for, ketika $i = 0$, dan $i < 9$, maka akan menjalankan $j = 0$ dan $j < 1$, maka program akan dieksekusi, dan akan menampilkan nilai 1. Begitu seterusnya looping berjalan hingga count bernilai 10 dan $i = 9$ dan $i < 9$, maka bernilai false, dan program akan berhenti.

5. Perulangan menggunakan while

```
WhileCount.java - Notepad
File Edit View

class WhileCount {
    public static void main (String args[]) {
        int count=1;
        int i=0;
        while (i<9) {
            int j=0;
            while (j<i+1) {
                System.out.print(count);
                j++;
            }
            count++;
            System.out.println();
            i++;
        }
    }
}
```

Menampilkan hasil keluaran dari program di atas.

```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>javac WhileCount.java

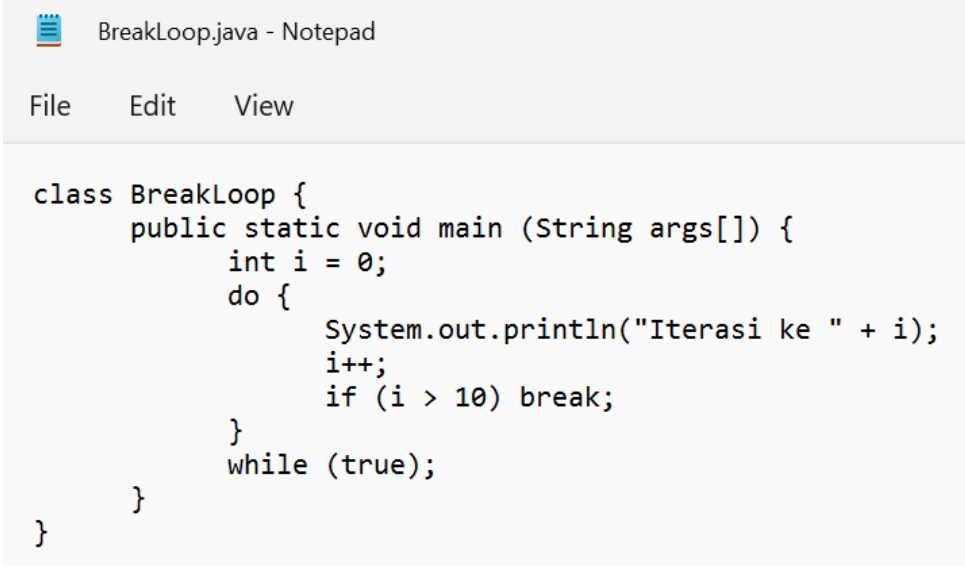
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java WhileCount.java
1
22
333
4444
55555
666666
7777777
88888888
999999999

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>_
```

Analisa :

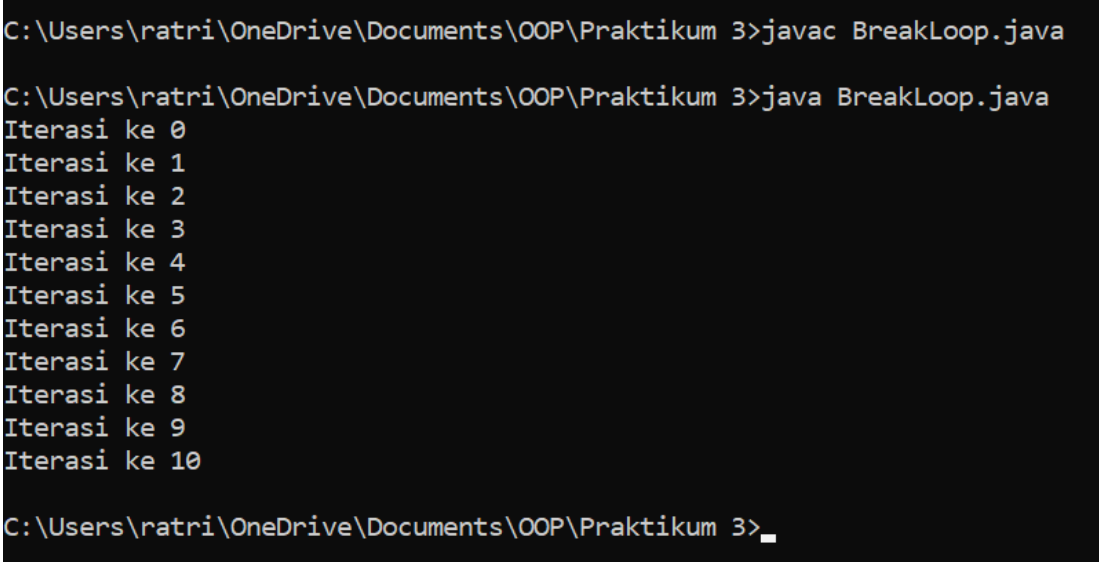
Variable count pada program di atas bernilai 1, dan variable i bernilai 0, maka perulangan menggunakan while akan dieksekusi ketika $i < 9$, variable $j = 0$ dan $j < 1$, maka akan menampilkan output 1. Selanjutnya increment terhadap variable i, j, count, ketika $i = 1$ dan $i < 9$, $j = 1$ dan $j < 2$, maka akan menampilkan output 2. Selanjutnya akan terjadi pengecekan terus menerus hingga variable count bernilai 10, variable $i = 9$ dan $i < 9$, maka bernilai false dan program berhenti.

6. Perulangan menggunakan break



```
class BreakLoop {
    public static void main (String args[]) {
        int i = 0;
        do {
            System.out.println("Iterasi ke " + i);
            i++;
            if (i > 10) break;
        }
        while (true);
    }
}
```

Menampilkan hasil keluaran dari program di atas



```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>javac BreakLoop.java
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java BreakLoop.java
Iterasi ke 0
Iterasi ke 1
Iterasi ke 2
Iterasi ke 3
Iterasi ke 4
Iterasi ke 5
Iterasi ke 6
Iterasi ke 7
Iterasi ke 8
Iterasi ke 9
Iterasi ke 10
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>_
```

Analisa :

Pada program di atas menggunakan statement do while. Pernyataan do akan dikerjakan terlebih dahulu kemudian akan dilakukan pengecekan terhadap kondisi while, apabila kondisi while true, maka program akan di eksekusi. Variable i bernilai 0, kemudian dilakukan perulangan, pada pernyataan yang terdapat dalam do, akan mencetak “iterasi ke 0.....iterasi ke 10” terdapat increment terhadap variable i hingga bernilai 10, namun jika variable $i > 10$ maka akan dilakukan pengecekan terhadap while, dan jika while bernilai false, maka akan dilakukan break, dan keluar dari perulangan tersebut.

7. Perulangan menggunakan continue

```
ContinueLoop.java - Notepad
File Edit View

public class ContinueLoop {
    public static void main(String args[]) {
        int a, b;
        for(a=0;a<2;a++)
            for(b=0;b<3;b++) {
                if (b==1) continue;
                System.out.println("a=" + a + " ; b=" + b);
            }
    }
}
```

Menampilkan hasil keluaran dari program di atas

```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>javac ContinueLoop.java

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java ContinueLoop.java
a=0 ; b=0
a=0 ; b=2
a=1 ; b=0
a=1 ; b=2

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>_
```

Analisa :

Nilai $a = 0$ dan $0 < 2$, nilai $b = 0$ dan $0 < 3$, maka akan mencetak nilai yang ada pada variable $a = 0$ dan nilai yang ada pada variable $b = 0$. Selanjutnya apabila nilai $a = 1$ dan $1 < 2$, nilai $b = 1$ dan $1 < 3$, maka akan mencetak nilai yang ada pada variable $a = 1$ dan nilai pada variable $b = 0$, karena terdapat ekspresi **if (b==1) continue**, yang berarti ketika variable b bernilai 1, maka akan menyebabkan suatu kondisi untuk melanjut ke tahapan selanjutnya pada perulangan, dan melewati nilai 1.

8. Pemakaian label pada kondisi break

```
BreakLabel.java - Notepad
File Edit View

public class BreakLabel {
    public static void main(String args[]) {
        int a, b;
        Mulai:
        for(a=0;a<2;a++)
            for(b=0;b<3;b++) {
                if (b==1) break Mulai;
                System.out.println("a=" + a + " ; b=" + b);
            }
    }
}
```


Menampilkan hasil keluaran dari program di atas.

```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>javac BreakLabel.java

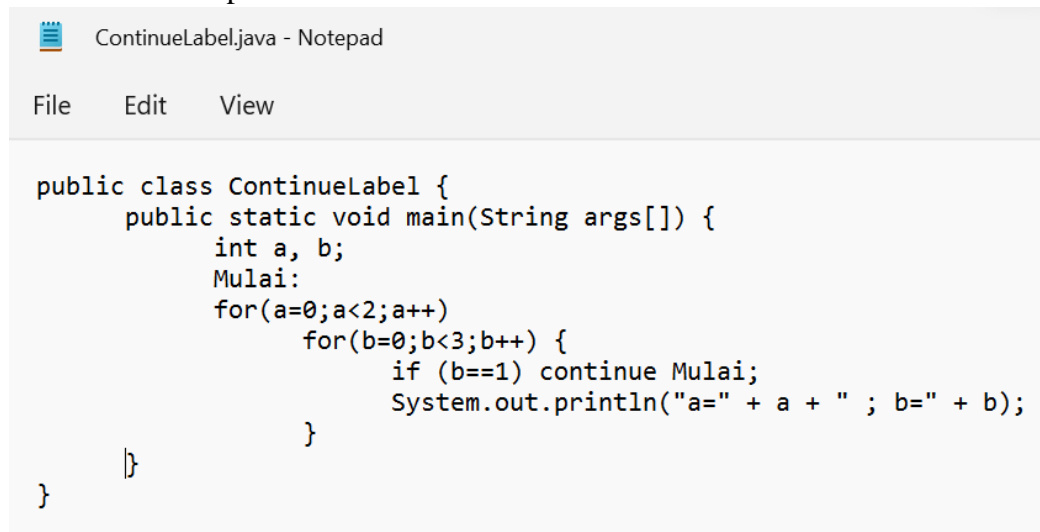
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java BreakLabel.java
a=0 ; b=0

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>_
```

Analisa :

Ketika pernyataan break label ditemukan selama iterasi, a = 0, dan b = 1, memutus looping dan melewati eksekusi pernyataan print dalam loop, dan program berakhir.

9. Pemakaian label pada kondisi continue



```
ContinueLabel.java - Notepad

File Edit View

public class ContinueLabel {
    public static void main(String args[]) {
        int a, b;
        Mulai:
        for(a=0; a<2; a++)
            for(b=0; b<3; b++) {
                if (b==1) continue Mulai;
                System.out.println("a=" + a + " ; b=" + b);
            }
    }
}
```

Menampilkan hasil keluaran dari program di atas

```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>javac ContinueLabel.java

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java ContinueLabel.java
a=0 ; b=0
a=1 ; b=0

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>
```

Analisa :

C. LATIHAN

1. Menampilkan bilangan factorial

```
faktorial.java - Notepad
File Edit View

import java.util.Scanner;
public class faktorial {
    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        System.out.print("Jumlah bilangan : ");

        int bil=scan.nextInt(), total=1;

        System.out.println("\n n!");
        System.out.println("-----");

        for (int i = 1; i <= bil; i++) {
            total*=i;
            System.out.println(i+" "+total);
        }
    }
}
```

Menampilkan hasil keluaran dari program di atas

```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>javac faktorial.java

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java faktorial.java
Jumlah bilangan : 7
n n!
-----
1 1
2 2
3 6
4 24
5 120
6 720
7 5040

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>_
```

Analisa :

Program diatas digunakan untuk menampilkan bilangan factorial, dengan menggunakan **import java.util.Scanner** untuk melakukan penginputan. **Scanner scan = new Scanner(System.in)** digunakan untuk membuat instansi scanner baru yang menunjuk ke aliran input yang diteruskan sebagai argument.

2. Menampilkan deret bilangan genap kecuali kelipatan 6

```
bilGenap.java - Notepad
File Edit View

public class bilGenap{
    public static void main(String[] args){
        for (int i = 2; i <= 20; i+=2){
            if(i%6 != 0){
                System.out.print(i + " ");
            }
        }
        System.out.println("");
    }
}
```

Menampilkan hasil keluaran dari program di atas.

```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>javac bilGenap.java

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java bilGenap.java
2 4 8 10 14 16 20

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>_
```

Analisa :

Untuk menampilkan bilangan genap dari 2 sampai 20, variable i saya isikan dengan nilai 2 terlebih dahulu, selanjutnya akan dilakukan pengecekan apabila nilai ($i \% 2 \neq 0$), maka akan dicetak, dilakukan perulangan secara terus menerus apabila nilai pada variable $i \leq 20$, dan dilakukan increment terhadap variable tersebut, jadi nilai i akan bertambah dua nilai setelah operasi tersebut di proses.

D. TUGAS

1. Deret Fibonacci



```
fibonacci.java - Notepad
File Edit View

import java.util.Scanner;
public class fibonacci{
    public static void main(String[] args){

        Scanner scan = new Scanner(System.in);

        int bil, hasil;
        int f1=0, f2=1;

        System.out.print("Masukkan bilangan ");
        bil = scan.nextInt();

        System.out.println("\n Bilangan fibonacci dari " + bil);
        System.out.print(" = 0, 1");

        while(true){
            hasil = f2 + f1;
            if(hasil > bil){
                break;
            }

            System.out.print(", " + hasil);
            f1 = f2;
            f2 = hasil;
        }
    }
}
```

Menampilkan hasil keluaran dari program di atas

```
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>javac fibonacci.java

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>java fibonacci.java
Masukkan bilangan 8

Bilangan fibonacci dari 8
= 0, 1, 1, 2, 3, 5, 8
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 3>_
```

